Approximate Analysis of Non-stationary Loss Queues and Networks of Loss Queues with General Service Time Distributions

N. Izady[a,*], D. Worthington[a],

[a]*Management Science Department, Lancaster University, Lancaster LA1 4YX, United Kingdom*

Abstract

A fixed point approximation (FPA) method has recently been suggested for non-stationary analysis of loss queues and networks of loss queues with Exponential service times. Deriving exact equations relating time-dependent mean numbers of busy servers to blocking probabilities, we generalize the FPA method to loss systems with general service time distributions. These equations are combined with associated formulae for stationary analysis of loss systems in steady state through a carried load to offered load transformation. The accuracy and speed of the generalized methods are illustrated through a wide set of examples.

Keywords: Queueing, Erlang loss model, Time-dependent arrival rate, Carried load

1. Introduction

In this paper, we shall be primarily concerned with time-dependent behavior of non-stationary loss queues denoted by $M_t/GI/s/0$. The arrival stream is assumed to be a non-homogeneous Poisson process (the M_t) with deterministic arrival rate function $\{\lambda(t),\, t \geq 0\}$. Service times are independent identically distributed (i.i.d) random variables following a general distribution (the GI) that are also independent of the arrival process. There is no extra waiting space (the 0), so customers finding all s parallel identical servers busy will be cleared from the system without affecting future arrivals (no retrials).

Let $Q(t)$ denote the number of busy servers in the system at time t. The *blocking probability function*, defined by $\beta(t) \equiv \Pr\{Q(t) = s\}$, is the most important per-

*Corresponding Author. Tel: +44(0)1524592721

Email addresses: n.izady@lancaster.ac.uk (N. Izady), d.worthington@lancaster.ac.uk (D. Worthington)

formance indicator of the system. Due to the independent increments property of the Poisson process, apart from the time-dependent probability of all servers being busy, $\beta(t)$ also represents the conditional probability of blocking at instant t given that an arrival occurs at t (Massey and Whitt, 1996):

$$\beta(t) \equiv \Pr\{Q(t) = s\} = \Pr\{Q(t) = s | \text{ an arrival occurs in } (t, t + dt)\}. \tag{1}$$

The steady-state analysis of the stationary version of the above system, the $M/GI/s/0$ queue, with constant arrival rate $\lambda(t) = \lambda$ and mean service time $1/\mu$ gives rise to the following equation (see, e.g. Gross and Harris, 1998, Chapter 5)

$$\lim_{t\to\infty} \Pr\{Q(t) = i\} = \frac{r^i/i!}{\sum_{j=0}^{s} r^j/j!}, \quad i = 0, \ldots, s, \tag{2}$$

where $r = \lambda/\mu$ is the so-called *offered load*. The offered load r coincides with the steady-state mean number of busy servers in an associated $M/GI/\infty$ model, with the same arrival and service processes as the loss system but with infinitely many servers. Substituting $i = s$ in (2) results in the well-known *Erlang Loss Equation* for computing the steady-state blocking probability:

$$\beta \equiv \lim_{t\to\infty} \Pr\{Q(t) = s\} = \frac{r^s/s!}{\sum_{j=0}^{s} r^j/j!}. \tag{3}$$

It then easily follows that

$$m \equiv \lim_{t\to\infty} E[Q(t)] = r(1 - \beta). \tag{4}$$

The mean number m of busy servers is referred to as *the carried load*, which, in contrast with the offered load, reflects losses in the workload due to lack of enough servers.

Loss queues and networks of loss queues have been used for modeling a wide range of systems from computer and telecommunication networks (e.g. Jagerman, 1975; Jennings and Massey, 1997; Abdalla and Boucherie, 2002; Alnowibet and Perros, 2006) to hospital wards (e.g. Bekker and Bruin, 2009; Bruin et al., 2009). The majority of these systems have significant variations in their arrival rates. This renders loss systems difficult to analyze. The insensitivity property of the system performance to the service time distribution beyond its mean, as in (2), is observed to be lost in this case (Davis et al., 1995). If service time follows a phase-type distri-

bution, one can always use a numerical ordinary differential equation (ODE) solver, like a *Runge-Kutta* (Green et al., 1991) or *Euler* method (Davis et al., 1995), or a faster approach like *randomization* (Ingolfsson et al., 2007) to compute $\Pr\{Q(t) = i\}$ over time. Since phase-type distributions are dense in the class of all distributions defined on non-negative real numbers (Asmussen, 2003, Theorem 4.2), one can in theory match empirical service time data sufficiently closely using a sufficiently large number of phases .

However, the computational effort associated with these numerical methods grows exponentially with the number of service time phases. For example, the size of the state space of a $100-$server loss system grows from 101 to 5152 when a two-phase distribution is considered instead of an Exponential. The curse of dimensionality is more profound when multiple classes of customers and networks of loss queues are considered. We therefore need to resort to approximate approaches.

Most approximate approaches treat the non-stationary $M_t/GI/s/0$ model at time t as if it were a stationary $M/GI/s/0$ model in steady state with $r = r(t)$, a properly defined time-dependent offered load. For example, the *Pointwise Stationary Approximation* (PSA) defines $r(t)$ as the instantaneous load $\lambda(t)/\mu$ (Green and Kolesar, 1991; Whitt, 1991). This approach implies that the system settles down to steady state at each time t, which does not generally occur unless the arrival rates change slowly relative to the mean service time. It also does not capture the effect of service time distribution beyond its mean.

Analogous to what happens in stationary loss systems, the *Modified Offered Load* (MOL) approximation defines $r(t)$ as the mean number of busy servers in an uncapacitated $M_t/GI/\infty$ model, with the same arrival and service processes as the original loss model, at any time t (Jagerman, 1975). It works well as long as blocking probabilities are small (Massey and Whitt, 1994).

The *Fixed Point Approximation* (FPA), proposed by Alnowibet and Perros (2009), takes a more dynamic approach to defining $r(t)$ and produces remarkable accuracy for the whole range of blocking probabilities in a short time. However, its application is limited to Exponential service times. It is based upon the following differential equation in $M_t/M/s/0$ loss queues

$$\frac{d\mathrm{E}[Q(t)]}{dt} = \lambda(t)(1 - \beta(t)) - \mu\mathrm{E}[Q(t)], \tag{5}$$

where μ is the Exponential service rate. The FPA method uses Equation (5) along with the Erlang loss equation (3) in an iterative manner to approximate blocking

probabilities and mean busy servers over time. To make use of the Erlang loss equation, Alnowibet and Perros (2009) propose the offered load $r(t)$ to be defined as

$$r(t) = \frac{\mathrm{E}[Q(t)]}{1 - \beta(t)}. \tag{6}$$

In this paper, we extend the FPA method to loss queues with arbitrary service time distributions. This is achieved by developing an integral equation relating the time-dependent mean number of busy servers (the carried load) to the blocking probabilities in $M_t/GI/s/0$ loss models. This equation is obtained by applying a decomposition technique to non-stationary infinite-server queues and gives rise to an exact algorithm for computing blocking probabilities in single-server loss queues. For multi-server loss queues, it replaces the differential equation (5) in the FPA iterative scheme.

The *carried load to offered load transformation* summarized in (6) is motivated by the similar relation between these two quantities in steady state, as illustrated in (4). We therefore investigate the accuracy of this transformation in non-stationary settings. Experiments confirm that the offered load obtained by this transformation can also be used for characterizing the entire distribution of the number of busy servers.

As shown by Alnowibet and Perros (2009), one can derive differential equations similar to (5) for multi-class loss queues and for networks of loss queues with Exponential service times. Combining those equations with associated stationary formulae, they extended the FPA method to multi-class loss queues and networks of loss queues. We do the same by generalizing our integral equation to cover multi-class loss queues and networks of loss queues with general service time distributions.

We start with single-class loss queues in Section 2. The derivations of an integral equation relating system characteristics, developing solution algorithms, and numerical experiments are included in this section. We then extend our method to multi-class loss queues and networks of loss queues in Sections 3 and 4, respectively. Conclusions are drawn in Section 5. This paper is accompanied by two online appendices. Appendix A includes all the required algorithms, and Appendix B provides an accuracy analysis for blocking probabilities computed for single-class loss queues.

2. Single-Class Loss Queues

In this section, we decompose a non-stationary $M_t/GI/\infty$ queue to derive an equation which expresses mean numbers of busy servers in $M_t/GI/s/0$ loss queues

in terms of the arrival rate, service time distribution, and blocking probability functions. For this purpose, we first need to review some results concerning non-stationary infinite-server queues.

Consider an $M_t/GI/\infty$ queue with a non-homogeneous Poisson arrival process with arrival rate function $\{\lambda(t), -\infty < t < \infty\}$. Let S be a generic service time random variable with cumulative distribution function (cdf) $G(t) \equiv \Pr(S \le t)$, $t \ge 0$. Let $Q_\infty(t)$ be the number of busy servers in the system at time t, and let $m_\infty(t) \equiv E[Q_\infty(t)]$. We assume the system starts empty in the distant past.

Theorem 1. *$Q_\infty(t)$ has a Poisson distribution for each value of t with the following time-dependent mean function*

$$m_\infty(t) = \int_{-\infty}^{t} \lambda(u) G^c(t-u)\, du. \tag{7}$$

The departure process is a Poisson process with the following time-dependent rate function

$$\delta_\infty(t) = \int_{0}^{\infty} \lambda(t-u) dG(u). \tag{8}$$

Proof. See Theorem 1 of Eick et al. (1993). □

Remark 1. Equation (7) remains valid even with a general arrival process provided that the time-dependent arrival rate function $\lambda(t)$ is well defined (Massey and Whitt, 1993, Theroem 2.1 and Remark 2.3). The queue length distribution, however, will not be Poisson any more.

Now consider an $M_t/GI/s/0$ loss system with the same arrival process and service time distribution as defined above for the infinite-server system. Let $Q(t)$ denote the number of busy servers at time t, and let $m(t) \equiv E[Q(t)]$. Let $\delta(t)$ denote the departure rate at time t.

Theorem 2. *For $M_t/GI/s/0$ loss system that starts out empty in the infinite past, we have*

$$m(t) = \int_{-\infty}^{t} \lambda(u)(1-\beta(u)) G^c(t-u)\, du, \tag{9}$$

and

$$\delta(t) = \int_{0}^{\infty} \lambda(t-u)(1-\beta(t-u)) dG(u), \tag{10}$$

where $\beta(t) = Pr\{Q(t) = s\}$.

Proof. Suppose we have an infinite-server system whose servers are numbered arbitrarily $1, 2, \ldots$ (see Figure 1). Decompose this system into an $s-$server *primary group* (numbered $1, 2, \cdots, s$) and an infinite-server *overflow group* (numbered $s+1, \cdots$). For overflow systems associated with stationary loss systems in steady state, see, for example, Chapter 7 of Wolff (1988).

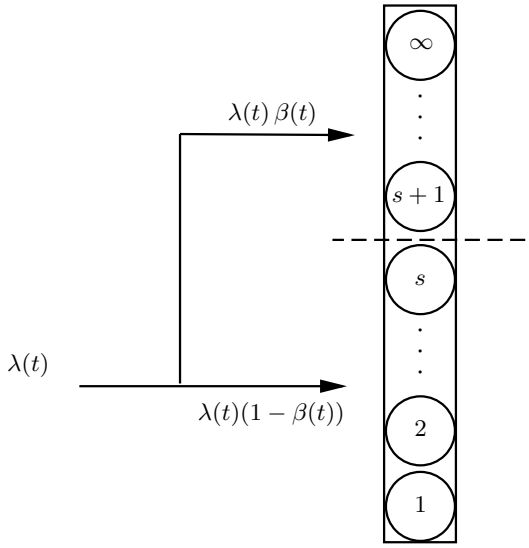

Figure 1: Primary group and overflow group

Now suppose that arrivals, which follow a non-homogeneous Poisson process with rate $\lambda(t)$, first refer to the primary group for service and start their service if an idle server is available there. Those who find all s servers in the primary group busy are not turned away, but overflow and are handled by the infinite-server overflow group. So the arrival stream is split into two substreams, one goes into the primary group, and the other is served in the overflow group. Note that the service time of the overflow group is the same as that of the primary group.

Now let $Q(t)$, $Q_o(t)$, and $Q_\infty(t)$ denote the number of busy servers in the primary $s-$server group, the infinite-server overflow group, and the entire system, respectively. We have

$$Q_\infty(t) = Q(t) + Q_o(t), \tag{11}$$

and by taking expectations

$$m(t) = m_\infty(t) - m_0(t), \tag{12}$$

where $m(t) = \mathrm{E}[Q(t)]$, $m_o(t) = \mathrm{E}[Q_o(t)]$, and $m_\infty(t) = \mathrm{E}[Q_\infty(t)]$. Since the arrival process to the system is a non-homogeneous Poisson process with rate $\lambda(t)$, we

immediately have $m_\infty(t)$ from (7). On the other hand, the arrival process to the overflow group is not Poisson. This is because overflows only occur when the primary group is full, and so the arrivals to the overflow group are Poisson only for the intervals during which this is true. However, Remark 1 allows us to use Equation (7) given a well-defined rate function exists for the overflow arrival process. In order to find this rate, we use a simple partitioning argument: Partition the interval $(-\infty, t)$ into sub-intervals of length δu and let $A(t)$ denote the total number of overflows in $(-\infty, t)$. Since the orderliness property of the Poisson process is preserved, the expected number of overflows over the sub-interval $(u, u + du)$ is $\lambda(u)\beta(u)du$ and so

$$E[A(t)] = \int_{-\infty}^{t} \lambda(u)\beta(u)du, \tag{13}$$

which implies that $\lambda(t)\beta(t)$ is the arrival rate to the overflow group. Now, we have

$$\begin{aligned} m(t) &= \int_{-\infty}^{t} \lambda(u)G^c(t-u)\,du - \int_{-\infty}^{t} \lambda(u)\beta(u)G^c(t-u)\,du \\ &= \int_{-\infty}^{t} \lambda(u)(1-\beta(u))G^c(t-u)\,du, \end{aligned} \tag{14}$$

which establishes (9).

Now let $D(t)$, $D_o(t)$, and $D_\infty(t)$ denote the total number of departures from the primary $s-$ server group, the overflow group, and the entire system, respectively, up to time t. We then have

$$E[D_\infty(t)] = \int_{-\infty}^{t} \lambda(u)du - \int_{-\infty}^{t} \lambda(u)G^c(t-u)\,du \tag{15}$$

$$E[D_o(t)] = \int_{-\infty}^{t} \lambda(u)\beta(u)du - \int_{-\infty}^{t} \lambda(u)\beta(u)G^c(t-u)\,du, \tag{16}$$

where the first terms on the right hand sides are the expected numbers of arrivals up to time t, and the second terms are the mean numbers of busy servers at time t. Thus,

$$E[D(t)] = E[D_\infty(t)] - E[D_o(t)] = \int_{-\infty}^{t} \lambda(u)(1-\beta(u))G(t-u)du, \tag{17}$$

which, by taking derivative with respect to t, yields the departure rate $\delta(t)$ given in (10). □

The following corollary states the relation between arrival and departure rates

in non-stationary loss queues.

Corollary 1.

$$\frac{dm(t)}{dt} = \lambda(t)(1 - \beta(t)) - \delta(t), \tag{18}$$

Proof. Proof follows easily by differentiating Equation (9) with respect to t. □

For $M_t/M/s/0$ loss queues with Exponential service times with mean $1/\mu$, we have from (9) and (10) that $\delta(t) = m(t)\mu$. Replacing this in Corollary 1 yields the differential equation (5) used in the FPA method for performance evaluation of $M_t/M/s/0$ loss queues.

We now develop algorithms for computing blocking probabilities and mean busy servers by virtue of Equation (9). We assume that the loss system starts empty at $t = 0$, which is equivalent to setting $\lambda(t) = 0$ for $t < 0$ in Theorem 2. We have not been able to analyze other initial settings, but with an appropriate choice of origin this covers many practical situations.

2.1. Single-server Loss Queues

For the $M_t/GI/1/0$ loss system, $m(t) = \beta(t)$. Hence, Equation (9) becomes

$$\beta(t) = \int_0^t \lambda(u)(1 - \beta(u))G^c(t - u)du. \tag{19}$$

Whilst it would be possible to produce a closed form solution for Equation (19) for some sufficiently simple arrival rate functions and service time distributions, in general we have to use numerical approaches. To do so, we subdivide the interval of integration $(0, t)$ into n equal subintervals with length $h = t/n$, and employ *the trapezoidal* rule of integration to obtain

$$\beta(t) = h[\lambda(0)(1 - \beta(0))G^c(t) + 2\sum_{i=1}^{n-1} \lambda(ih)(1 - \beta(ih))G^c(t - ih) + \lambda(t)]/(2 + h\lambda(t)). \tag{20}$$

Assuming $\beta(0) = 0$, which is an appropriate assumption when the system starts empty, we will be able to work out blocking probabilities at desired points of time in a sequential manner. This approach is outlined in Algorithm 1 in Appendix A of the online supplement. Apart from inevitable numerical error produced by approximating the integral with an equivalent summation, Algorithm 1 is exact.

2.2. Multi-server Loss Queues

Since Expression (9) contains two unknown functions, $\beta(t)$ and $m(t)$, for multi-server loss queues another equation relating these two functions is needed. The Erlang loss formula seems an appropriate complementary equation given a sensible definition of time-dependent offered load $r(t)$. In line with Alnowibet and Perros (2009), we define

$$r(t) = m(t)/(1 - \beta(t)), \tag{21}$$

and use

$$\beta(t) = \frac{r(t)^s/s!}{\sum_{i=0}^{s} r(t)^i/i!}, \tag{22}$$

as an approximate complementary equation. In fact, Equation (22) is exact for single-server loss queues, as can bee seen by setting $s = 1$ and $m(t) = \beta(t)$. Equations (9), (21), and (22) can now be solved iteratively as follows:

1. Choose an appropriate tolerance ϵ, step size h, and final time T.
2. Start with initial value $\beta^0(t) = 0.0$ for all $0 \leq t \leq T$.
3. Set the iteration counter $k = 0$.
4. Calculate $m(t) = \int_0^t \lambda(u)(1 - \beta^k(u))G^c(t - u)du$ for all $0 \leq t \leq T$.
5. Calculate $r(t) = m(t)/(1 - \beta^k(t))$ for all $0 \leq t \leq T$.
6. Update the blocking probabilities: $\beta^{k+1}(t) = \frac{r(t)^s/s!}{\sum_{i=0}^{s} r(t)^i/i!}$ for all $0 \leq t \leq T$.
7. If $\max_{0\leq t\leq T}\{|\beta^{k+1}(t) - \beta^k(t)|\} < \epsilon$, then return $\beta^{k+1}(t)$ for all $0 \leq t \leq T$ and stop; otherwise set $k = k + 1$, and return to Step 4.

Remark 2. It follows upon setting $\beta^0(t) = 0$, $0 \leq t \leq T$, that $m(t)$ obtained in Step 4 above equals $m_\infty(t)$ defined in (7), and so $r(t) = m_\infty(t)$. Hence, the first iteration of the above routine produces the same values for blocking probabilities as the MOL approach.

The structure of this approach is similar to that of the FPA method; at each iteration, it computes blocking probabilities at all epochs based on the estimates of blocking probabilities obtained in the previous iteration. It turns out to be more efficient, especially with large numbers of servers, to work out the 'accurate' estimate of the blocking probability at each point before proceeding to the next point. This approach is illustrated in Algorithm 2 in Appendix A of the online supplement. It uses the trapezoidal integrating method and calculates blocking probabilities by a recursive formula to make computations numerically stable.

2.3. Numerical Results

In this section, we investigate the accuracy and speed of Algorithms 1 and 2 across various service time distributions and with different numbers of servers. Like other papers in this context, we use a sinusoidal arrival rate function $\lambda(t) = \bar{\lambda}(1 + \alpha \sin(2\pi t/C))$, where $\bar{\lambda}$ is the average arrival rate, α is the relative amplitude, and C is the cycle time. We set $\alpha = 0.5$ and $C = 24$ hours in all test cases.

We carried out our experiments with Exponential, Hyper-Exponential, and Erlang-2 distributions as phase-type service times, and with Log-Normal distribution as a non-phase type service time. Three parameters, mean, SCV (variance divided by mean squared), and r (a measure reflecting the third moment) are needed for characterizing the Hyper-Exponential distribution as described in Davis et al. (1995). The Exponential and Erlang-2 distributions are completely defined by their mean values, and their SCVs equal 1.0 and 0.5, respectively. For the Log-Normal distribution, mean and SCV are needed.

Mean service times were assumed to be four hours in all cases. This relatively long service time was deliberately chosen so as to demonstrate the ability of proposed algorithms in coping with hard cases; both PSA and MOL tend to work better when service times are relatively short. Long service times are also common in healthcare delivery processes. For the Hyper-Exponential distribution, SCV was set to 4.0 and r to 0.1, 0.5, and 0.9. The SCV of the Log-Normal distribution was chosen to be 2.0.

We implemented our algorithms in MATLAB. For systems with phase-type service times, we used the 'ode45' function of the MATLAB ODE suit to generate exact results (Shampine and Reichelt, 1997). It numerically solves the Chapman-Kolmogorov differential equations describing the system dynamics using a Runge-Kutta method, and is widely used as a benchmark (Ingolfsson et al., 2007). For the Log-Normal distribution, simulation is the only available benchmark. We replicate the simulation model 10000 times to reach a high level of accuracy. We compute blocking probabilities at five minute intervals over a period of four days ($T = 96$ hours).

For single-server loss queues, we experimented with $\bar{\lambda} = 0.01, 0.1, 1.0$, and 5.0 to cover a wide range of blocking probabilities. Results for two extreme cases are plotted in Figure 2. As Equation (19) is exact for single-server systems, the only source of error is due to the numerical integration in Equation (20). As a consequence, excellent accuracy is observed in these plots.

For multi-server loss queues, we set $\bar{\lambda} = 35$ and experimented with 200, 150, 100,

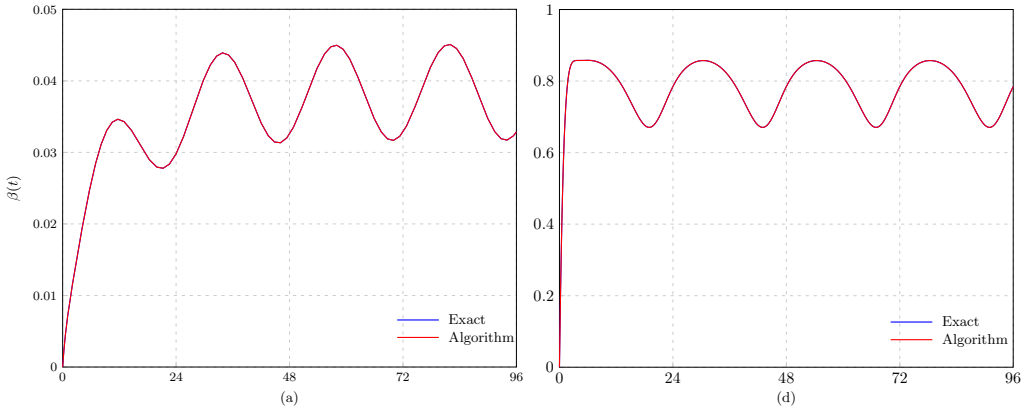

Figure 2: Blocking probability vs. time for single-server loss queues with (a) Hyper-Exponential service times with $r = 0.1$ and $\bar{\lambda} = 0.01$, and (b) Erlang-2 service times with $\bar{\lambda} = 5.0$.

and 50 servers to span a wide range of blocking probabilities and to test computation speed for large numbers of servers. We also implemented the MOL approach for calculating loss probabilities and have included its results for the sake of comparison. Tolerance of Algorithm 2 was set to 0.001 as accuracy more than three digits is unlikely to be required in practical situations. Nevertheless, smaller values can be chosen if needed.

A sample of results are illustrated in Figures 3, 4, and 5 for Hyper-Exponential with $r = 0.5$, Erlang-2, and Log-Normal distributions, respectively. It is observed in these plots that blocking probabilities obtained by Algorithm 2 (red lines) are always pretty close to the exact results (blue lines). It is also clear that the algorithm always performs more accurately than the MOL (green line). For 200 servers (part (a) of figures), the blocking probabilities are very small and the difference between MOL and the algorithm is not significant with both very close to exact results. However, as number of servers decreases to 150 (part (b) of figures) and blocking rises up to a peak around 20 percent, MOL deteriorates, underestimating the peaks, overestimating the troughs, and lagging behind the exact results. These problems of MOL become more serious for 100 (part (c) of figures) and 50 (part (d) of figures) servers while the proposed algorithm is working consistently well. The mean numbers of busy servers (not plotted) were also remarkably close to exact results for the proposed algorithm. Whilst not included here, results for 5-20 servers showed that the proposed algorithm also maintains its high accuracy levels for much smaller numbers of servers.

Relative and absolute accuracy of Algorithm 2 and MOL are reported in Appendix B of the online supplement for all the test cases used in this section. Results

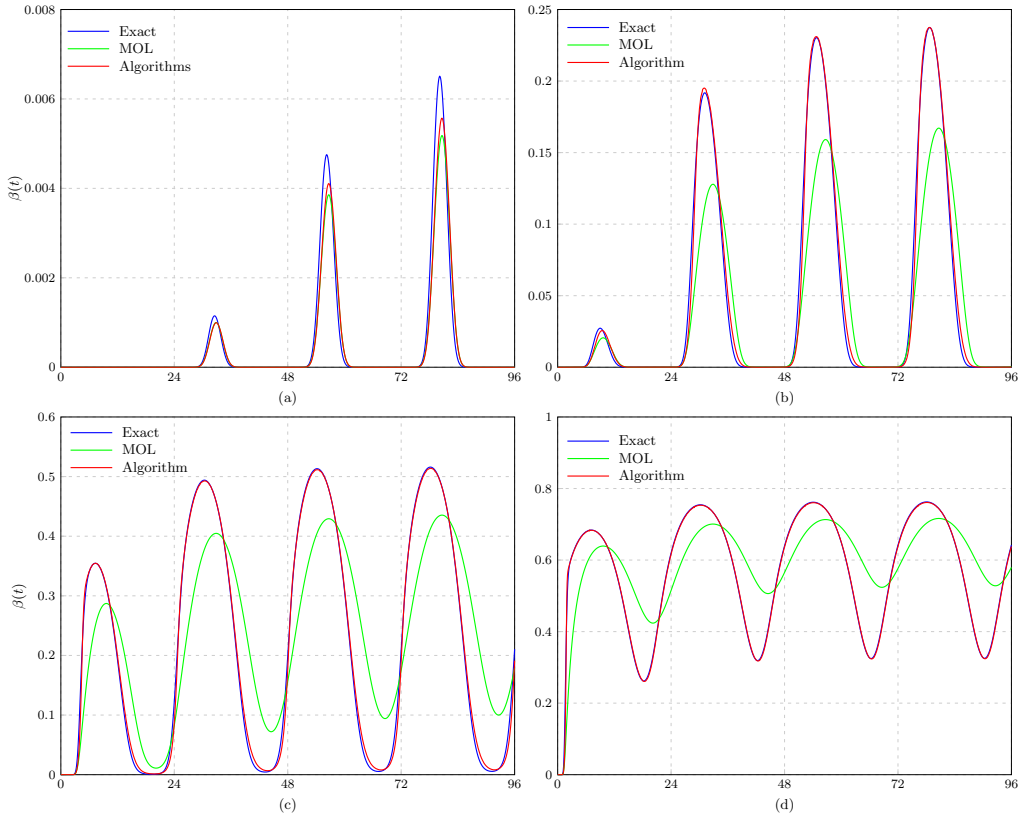

Figure 3: Blocking probability vs. time for loss queues with Hyper-Exponential service times with r= 0.5 and: (a) 200 servers, (b) 150 servers, (c)100 servers, and (d) 50 servers.

for the PSA (not included) also reveal significant errors. In particular, as it does not distinguish between service times beyond their means, the PSA produces identical results for Figures 3, 4, and 5, and also fails to show the transient behavior present in those figures.

The computation time of Algorithm 2 was less than 2 seconds in all test cases. The exact algorithm took over 3 minutes to produce results with 50 servers and more than 5 hours with 200 servers.

2.4. Non-stationary Erlang Loss Equation

As noted by Alnowibet and Perros (2009) for the special case of Exponential service times, the high degree of accuracy in the approximate results produced from equations (9), (21), and (22) suggests that Equation (22) with offered load defined in (21) might in fact be true for multi-server systems as it is for single-server systems. We therefore investigate this conjecture by comparing the values obtained from the

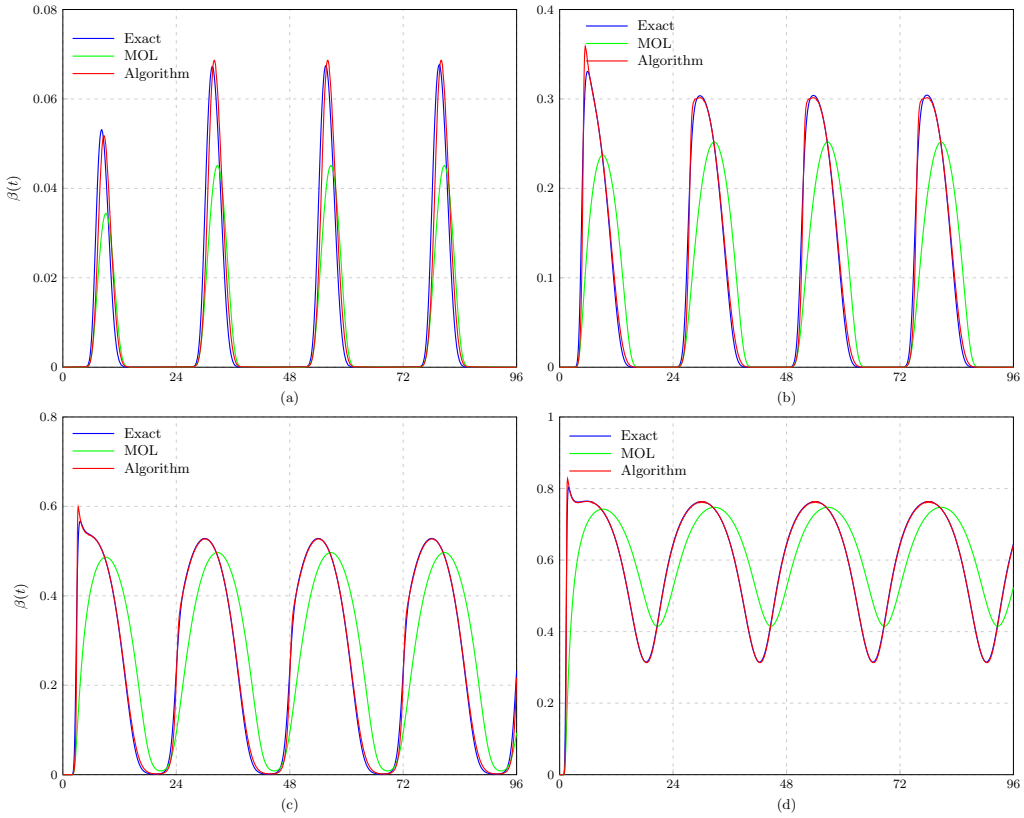

Figure 4: Blocking probability vs. time for loss queues with Erlang-2 service times and: (a) 200 servers, (b) 150 servers, (c) 100 servers, and (d) 50 servers.

right hand side and left hand side of the following equation

$$\beta(t) \simeq \frac{\left(\frac{m(t)}{1-\beta(t)}\right)^s / s!}{\sum_{i=0}^{s} \left(\frac{m(t)}{1-\beta(t)}\right)^i / i!}, \tag{23}$$

where both $\beta(t)$ and $m(t)$ are to be exact values computed by the Runge-Kutta method. If the values obtained from the two sides were equal, it would mean that the above equation is exact. Otherwise, it would be an approximate relation between $\beta(t)$ and $m(t)$.

We calculated the absolute difference between the right hand side and left hand side of Equation (23) at five minute intervals for all the test cases used in Section 2.3. A sample of results are plotted alongside associated absolute errors of Algorithm 2 in Figure 6. According to these plots, the absolute difference between the right hand side and left hand side of the above equation (labeled as *Erlang*) becomes as large as 0.01. Since we set a high level of accuracy for the Runge-Kutta method (six digits accuracy), these differences are not purely numerical errors. Notice also that

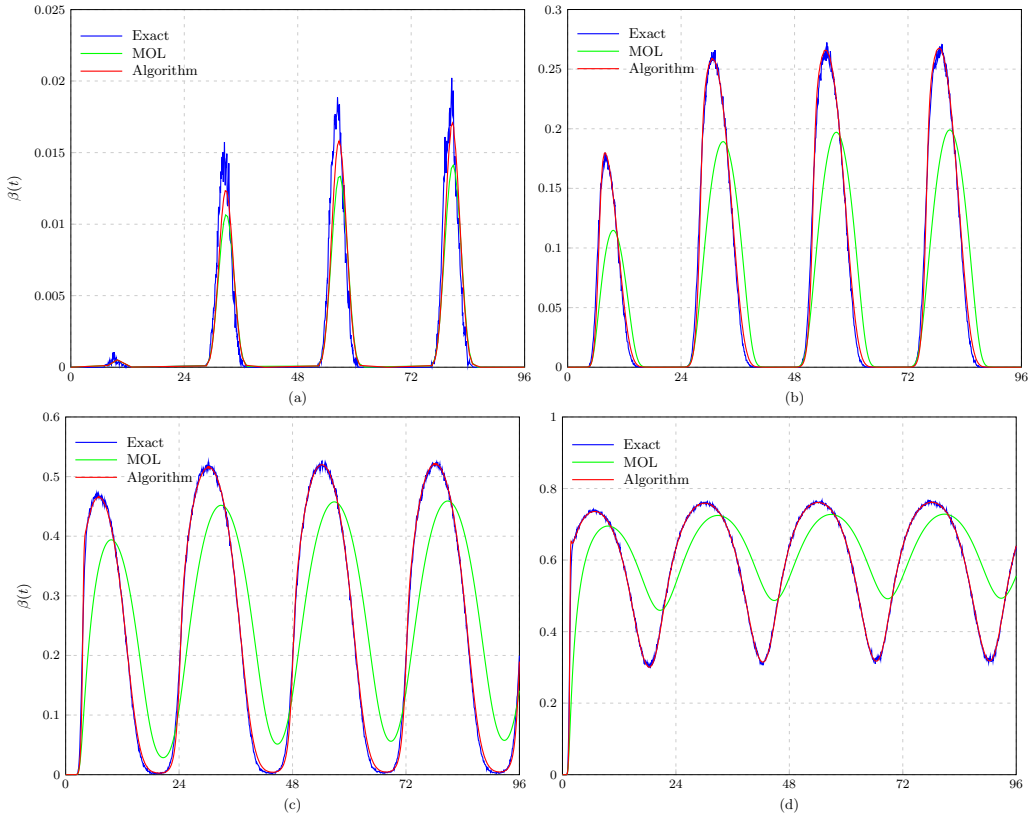

Figure 5: Blocking probability vs. time for loss queues with Log-Normal service times and: (a) 200 servers, (b) 150 servers, (c) 100 servers, and (d) 50 servers.

the absolute error of Algorithm 2 (labeled as *Algorithm*) follows the absolute error of the above equation pretty closely. These two observations support our conjecture that errors of Algorithm 2 stem from the approximate nature of the Erlang loss equation with $r(t) = m(t)/(1 - \beta(t))$ in non-stationary settings.

2.5. Queue Length Distribution

Theorem 2 tells nothing about the distribution of numbers of busy servers except for the easy case of $s = 1$. But, surprisingly, the corresponding stationary formula works well in time dependent cases with $r(t)$ defined in (21). Specifically,

$$\Pr\{Q(t) = i\} \approx \frac{\left(\frac{m(t)}{1-\beta(t)}\right)^i / i!}{\sum_{j=0}^{s} \left(\frac{m(t)}{1-\beta(t)}\right)^j / j!}, \quad i = 0, 1, \cdots, s. \tag{24}$$

We computed the queue length probability mass functions by substituting values of $m(t)$ and $\beta(t)$, obtained by Algorithm 2, into the above formula for all test cases of Section 2.3. This enabled us to estimate queue length distribution functions for all

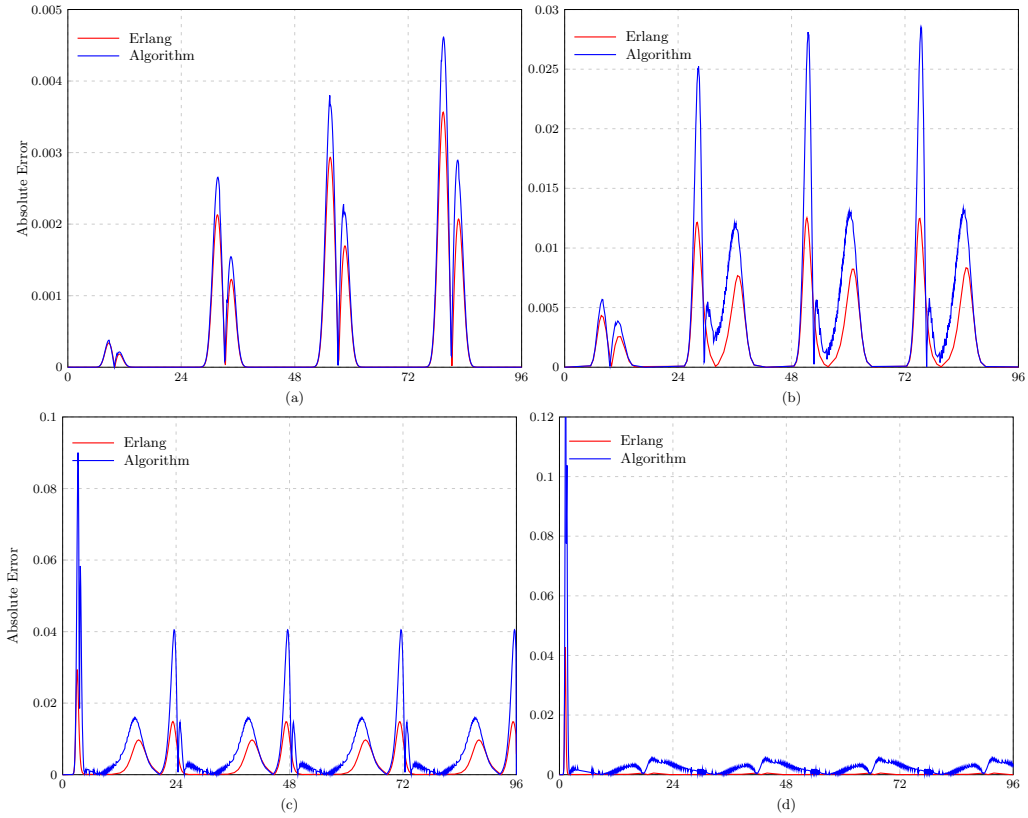

Figure 6: Absolute error vs. time for : (a) Hyper-Exponential with $r = 0.9$ and 200 servers, (b) Hyper-Exponential with $r = 0.5$ and 150 servers, (c) Exponential with 100 servers, and (d) Erlang-2 with 50 servers.

the test cases. To measure the accuracy of the estimated distribution functions, we used the *Kolmogrov-Smirnov* statistic defined as

$$\mathrm{KS}(t) = \max_{0 \le i \le s} |\tilde{F}_t(i) - F_t(i)|, \quad 0 \le t \le T, \tag{25}$$

where $\tilde{F}_t(i)$ and $F_t(i)$ are approximate and exact queue length distribution functions at time t obtained by Expression (24) and the Runge-Kutta method, respectively. In most of our test cases, the time average of $\mathrm{KS}(t)$ was less than 2 percent, and in all cases it was less than 4 percent. Figure 7 shows the mean, 10th percentile, and 90th percentile for the worst case (the 150-server example with Hyper-Exponential service times with $r = 0.1$), and nevertheless still shows a good match between approximate and exact results.

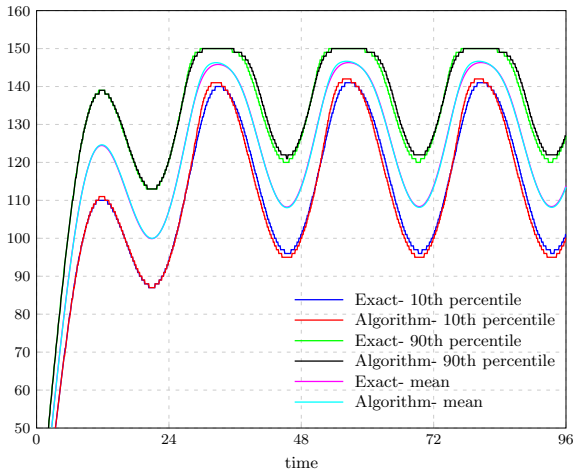

Figure 7: The mean, 10th percentile, and 90th percentile for the number of busy servers distribution.

3. Multi-Class Loss Queues

Consider an $s-$server loss system serving K independent classes of customers. A class k customer arrives to the system according to a non-homogeneous Poisson process with arrival rate function $\{\lambda_k(t),\ -\infty < t < \infty\}$ and requests for b_k servers for $k = 1, \ldots, K$. Given b_k servers are available, the arrival occupies them for a random amount of time distributed with the cdf function $G_k(t)$, $t \geq 0$. When the service is finished, all b_k servers are released simultaneously. An arriving customer is lost if the required servers are not all available. All servers are able to handle all customer classes. The arrivals and service times of each class of customers are assumed to be independent of each other and of other classes.

Let $Q_k(t)$ be the number of servers occupied by class k customers at time t, and let $m_k(t) \equiv E[Q_k(t)]$ for $k = 1, \ldots, K$. The blocking probability function for class k customers is defined as

$$\begin{aligned} \beta_k(t) \equiv & \Pr\{Q_k(t) > s - b_k\} \\ & = \Pr\{Q_k(t) > s - b_k \,|\, \text{a class } k \text{ arrival occurs in } (t, t+dt)\}, \end{aligned} \tag{26}$$

for $k = 1, \ldots, K$. The following corollary extends Theorem 2 to multi-class loss systems.

Corollary 2.

$$m_k(t) = b_k \int_{-\infty}^{t} \lambda_k(u)(1-\beta_k(u))G_k^c(t-u)du, \quad k = 1, \ldots, K, \tag{27}$$

$$\delta_k(t) = \int_{0}^{\infty} \lambda_k(t-u)(1-\beta_k(t-u))dG_k(u), \quad k = 1, \ldots, K. \tag{28}$$

Proof. Proof easily follows using the same decomposition method applied for Theorem 2. Since customers do not interact with each other in infinite-server systems, we can use Equation (7) to find mean busy servers in the entire system and in the overflow group for each class of customers independently of other classes. □

In order to approximate $\beta_k(t)$ and $m_k(t)$ for all customer classes, we combine Equation (27) with corresponding stationary formulae. Kaufman (1981) proposed an efficient method for evaluating the steady-state blocking probability β_k in a multi-class loss system where class k customers have constant arrival rate $\lambda_k(t) = \lambda_k$, $t \geq 0$, and arbitrary service time distributions with mean $1/\mu_k$ for $k = 1, \ldots, K$. Starting with $w_0 = 1$, this method computes w_j values recursively as follows:

$$w_j = \sum_{k=1}^{K} r_k w_{j-b_k}, \quad j = 1, \ldots, s, \tag{29}$$

where r_k is the workload brought to the system by class k customers and equals $b_k\lambda_k/\mu_k$. The blocking probability of class k customers is then evaluated as follows

$$\beta_k = \frac{\sum_{j=s-b_k}^{s} w_j}{\sum_{j=0}^{s} w_j}, \quad k = 1, \ldots, K. \tag{30}$$

For the non-stationary case, we replace r_k in Equation (29) with

$$r_k(t) = m_k(t)/(1-\beta_k(t)), \tag{31}$$

to obtain

$$w_j(t) = \sum_{k=1}^{K} r_k(t) w_{j-b_k}(t), \quad j = 1, \ldots, s, \tag{32}$$

and

$$\beta_k(t) = \frac{\sum_{j=s-b_k}^{s} w_j(t)}{\sum_{j=0}^{s} w_j(t)}, \quad k = 1, \ldots, K. \tag{33}$$

Solving Equations (27), (31), (32), and (33), iteratively, produces estimates of

blocking probabilities $\beta_k(t)$ and mean numbers $m_k(t)$ of busy servers occupied by class k customers for all k. The total number of busy servers would therefore be $m(t) = \sum_{k=1}^{K} m_k(t)$. The iterative process is outlined in Algorithm 3 in Appendix A of the Online Supplement.

As a numerical example, we considered a two-class loss system with $\lambda_1(t) = 35(1 + 0.5\sin(2\pi t/24))$ and $\lambda_2(t) = 20(1 + 0.5\sin(2\pi t/12))$. The first class customers require one server and their service time is assumed to follow a Log-Normal distribution with mean value of 4 hours and SCV of 2. The second class customers require 2 servers and their service time is a Log-Normal distribution with mean value of 2 hours and SCV of 4.0.

The blocking probability values for both classes of customers are plotted in Figure 8 alongside the corresponding values obtained by simulation experiments. The results show a high level of accuracy. The computation time was around 30 seconds with Algorithm 3, while the simulation model took more than 120 minutes to do 10000 replications.

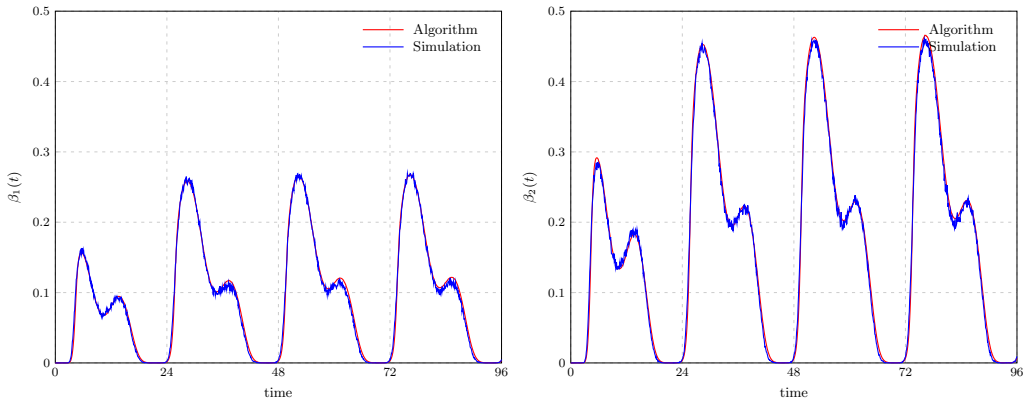

Figure 8: Blocking probability vs. time for the non-stationary multi-class example.

4. Networks of Loss Queues

Consider an $(M_t/GI/s_i/0)^I/M$ loss network with I service facilities, indexed by $i = 1, \ldots, I$, where facility i comprises s_i identical servers. The external arrival process to facility i, independent of other facilities, is a non-homogeneous Poisson process (the M_t) with deterministic arrival rate function $\{\lambda_i(t), -\infty < t < \infty\}$. Service times at facility i are assumed to be i.i.d random variables with an arbitrary cdf function $G_i(t)$ (the GI). The routing process is stationary Markovian (the M) with matrix $\boldsymbol{P} = [p_{ij}]$, where p_{ij} is the probability of a customer moving to facility j upon service completion in facility i; $q_i = 1 - \sum_{j=1}^{I} p_{ij}$ is therefore the probability

of a customer leaving the system after visiting facility i. The arrival, service, and routing processes are assumed to be mutually independent. An arriving customer (internal or external) to facility i finding all servers busy will be lost from the system.

Let $Q_i(t)$ be the number of busy servers in facility i, and let $m_i(t) \equiv E[Q_i(t)]$. The blocking probability function at facility i is defined as

$$\beta_i(t) = \Pr\{Q_i(t) = s_i | \text{an arrival (internal or external) occurs in } (t, t+dt)\}. \quad (34)$$

Notice that the blocking probability is no longer the same as the (unconditional) probability of all servers being busy. This is because the aggregate arrival process to each facility is not generally Poisson. We have the following theorem.

Theorem 3. *For the $(M_t/GI/s_i/0)^I/M$ loss system that starts out empty in the infinite past, we have*

$$m_i(t) = \int_{-\infty}^{t} \gamma_i(u) G_i^c(t-u)\, du, \quad (35)$$

where $\gamma_i(t)$ is the aggregate arrival rate function to facility i, defined as the minimal non-negative solution to the following system of input equations

$$\gamma_i(t) = \left[\lambda_i(t) + \sum_{j=1}^{I} p_{ji} \int_0^{\infty} \gamma_j(u) dG_j(t-u) du\right](1-\beta_i(t)), \quad i = 1, \ldots, I. \quad (36)$$

Proof. We prove the results by showing that mean busy servers in each facility of the $(M_t/GI/s_i/0)^I/M$ loss network is the same as mean busy servers in the corresponding facility of an associated infinite-server $(G_t/GI/\infty)^{I+1}/G_t$ network. Since expressions for mean busy servers of $(G_t/GI/\infty)^{I+1}/G_t$ systems are given in Massey and Whitt (1993), we can readily use them once the desired network characteristics are well defined.

We define an $(G_t/GI/\infty)^{I+1}/G_t$ network as follows. Suppose there are $I+1$ infinite-server facilities in the system, where the first I facilities have the same service processes as the corresponding facilities in the loss network, and the $(I+1)$th facility has an arbitrary service time distribution. The external arrival process to facility i is assumed to be a general arrival process with rate function

$$\lambda_i^{\infty}(t) = \begin{cases} \lambda_i(t)(1-\beta_i(t)), & i = 1, \ldots, I, \\ \sum_{i=1}^{I} \lambda_i(t)\beta_i(t), & i = I, \end{cases} \quad (37)$$

with $\beta_i(t)$ defined in (34). The routing process is assumed to be a general process with time-dependent transition matrix $\boldsymbol{P}^{\infty}(t) = [p_{ij}^{\infty}(t)]$, where

$$p_{ij}^{\infty}(t) = \begin{cases} p_{ij}(1-\beta_j(t)), & i,j = 1,\ldots,I, \\ \sum_{k=1}^{I} p_{ik}\beta_k(t), & i = 1,\ldots,I,\ j = I+1, \\ 0, & i = I+1. \end{cases} \tag{38}$$

With the above settings, the aggregate arrival process to facility i of the infinite-server network is the aggregate arrival process of customers admitted to facility i of the loss network for $i = 1,\ldots,I$. All the blocked customers, either internal or external, are placed in the $(I+1)$th facility, from which they leave the system after an arbitrary amount of time. The mean number of busy servers in the ith facility of the loss network is therefore the same as the mean number of busy servers in the ith facility of the constructed infinite-server network for $i = 1,\ldots,I$. Now, Equations (35) and (36) are readily obtained from $(G_t/GI/\infty)^{I+1}/G_t$ equations given in Theorem 1.2 of Massey and Whitt (1993) with the system parameters defined in (37) and (38). □

To develop a solution algorithm, suppose that the loss network starts empty at $t = 0$. To compute $m_i(t)$ and $\beta_i(t)$, in line with Alnowibet and Perros (2009), we define time dependent offered load at facility i as

$$r_i(t) = m_i(t)/(1-\beta_i(t)) \tag{39}$$

and use

$$\beta_i(t) = \frac{r_i(t)^{s_i}/s_i!}{\sum_{j=0}^{s_i} r_i(t)^j/j!}, \tag{40}$$

as the complementary equation for $i = 1,\ldots,I$. Equations (35), (36), (39), and (40) can now be solved iteratively, starting with initial values for blocking probabilities. In order to obtain $m_i(t)$ from (35), one needs to solve the system of input equations stated in (36) for $\gamma_i(t)$, $i = 1,\ldots,I$, numerically at each time t. To do so, we divide the interval $(0,t)$ into n equal sub-intervals of length $h = t/n$ and apply the

trapezoidal rule of integration to obtain

$$\left(\frac{2}{1-\beta_i(t)} - hp_{ii}g_i(0)\right)\gamma_i(t) - \sum_{j\neq i}^{I} hp_{ji}g_j(0)\gamma_j(t) = 2\lambda_i(t) + h\sum_{j=1}^{I} p_{ji}\gamma_j(0)g_j(t) + 2h\sum_{j=1}^{I}\sum_{k=1}^{n-1} p_{ji}\gamma_j(t-kh)g_j(kh), \quad i = 1,\ldots,I, \quad (41)$$

where $g_i(t)$ is the probability density function (pdf) associated with cdf $G_i(t)$. Writing the above in matrix form, we have

$$\boldsymbol{A}(t)\boldsymbol{\gamma}(t) = \boldsymbol{C}(t) + h\boldsymbol{P}^T\sum_{k=1}^{n-1}\boldsymbol{B}(kh)\boldsymbol{\gamma}(t-kh), \quad (42)$$

where $\boldsymbol{\gamma}(t) \equiv (\gamma_1(t),\ldots,\gamma_I(t))^T$, $\boldsymbol{A}(t) \equiv [a_{ij}(t)]$ is an $I \times I$ matrix with

$$a_{ij}(t) = \begin{cases} \frac{2}{1-\beta_i(t)} - hp_{ii}g_i(0), & \text{if } i = j, \\ hp_{ji}g_j(0), & \text{if } i \neq j, \end{cases} \quad (43)$$

$\boldsymbol{C}(t) \equiv (c_1(t),\ldots,c_I(t))^T$ with

$$c_i(t) = 2\lambda_i(t) + h\sum_{j=1}^{I} p_{ji}\gamma_j(0)g_j(t), \quad i = 1,\ldots,I, \quad (44)$$

and finally $\boldsymbol{B}(t) \equiv [b_{ij}(t)]$ is an $I \times I$ diagonal matrix with $b_{ii}(t) = g_i(t)$. Now, starting with $\gamma_i(0) = \lambda_i(0)$, $i = 1,\ldots,I$, and having estimates of $\beta_i(t)$, one can find $\boldsymbol{\gamma}(t)$ at desired points of time recursively as follows

$$\boldsymbol{\gamma}(t) = \boldsymbol{A}^{-1}\left(\boldsymbol{C}(t) + h\boldsymbol{P}^T\sum_{k=1}^{n-1}\boldsymbol{B}(kh)\boldsymbol{\gamma}(t-kh)\right) \quad (45)$$

The required steps of the iterative routine are outlined in Algorithm 4 in Appendix A of the online supplement.

As a numerical example, we consider a six-facility ($I = 6$) loss network with

following specifications:

$$
\begin{aligned}
&\lambda_1(t) = 8 + 6\sin(2\pi t/24), && S_1 \sim \text{Log-Normal}(2.0, 2), && s_1 = 20,\\
&\lambda_2(t) = 10 + 5\sin(2\pi t/12), && S_2 \sim \text{Log-Normal}(1.0, 2), && s_2 = 15,\\
&\lambda_3(t) = 25 + 20\sin(2\pi t/24), && S_3 \sim \text{Log-Normal}(0.75, 2), && s_3 = 20,\\
&\lambda_4(t) = 15 + 15\sin(2\pi t/48), && S_4 \sim \text{Log-Normal}(1.0, 2), && s_4 = 15,\\
&\lambda_5(t) = 20 + 10\sin(2\pi t/24), && S_5 \sim \text{Log-Normal}(0.5, 2), && s_5 = 10,\\
&\lambda_6(t) = 0, && S_6 \sim \text{Log-Normal}(1.5, 2), && s_6 = 20,
\end{aligned}
$$

where $S_i \sim \text{Log-Normal}(1/\mu_i, \sigma_i^2)$ denotes a Log-Normal service time in facility i with mean and variance of $1/\mu_i$ and σ_i^2, respectively. The routing matrix is as follows

$$
\boldsymbol{P} = \begin{bmatrix}
0 & 0.4 & 0.4 & 0 & 0 & 0\\
0 & 0 & 0 & 0.5 & 0.4 & 0\\
0.3 & 0 & 0 & 0.3 & 0.3 & 0\\
0 & 0 & 0.7 & 0 & 0 & 0.3\\
0 & 0.6 & 0 & 0 & 0 & 0.3\\
0.5 & 0 & 0 & 0 & 0.3 & 0
\end{bmatrix} \tag{46}
$$

The results of running Algorithm 4 for the above example for 96 hours have been plotted in Figure 9. Compared to simulation results, obtained by 20000 replications, a high level of accuracy is observed. The lowest relative accuracy seem to be for the sixth facility in which blocking probabilities are small. The computation time with the simulation model was more than 10 hours, whereas it was thirty seconds for Algorithm 4.

5. Conclusion

By establishing integral equations relating time-dependent mean busy servers and blocking probabilities for non-stationary single-class, multi-class, and networks of loss queues with arbitrary service time distributions, we have extended the FPA method of Alnowibet and Perros (2009) substantially beyond the Exponential service time assumption. This generalized FPA method is shown to provide highly accurate results for a wide range of cases, including important cases where previous well-established methods such as PSA and MOL are known to perform poorly.

In comparison to exact methods such as the Runge-Kutta ODE solver, the generalized FPA method applies to all service time distributions, performs much faster,

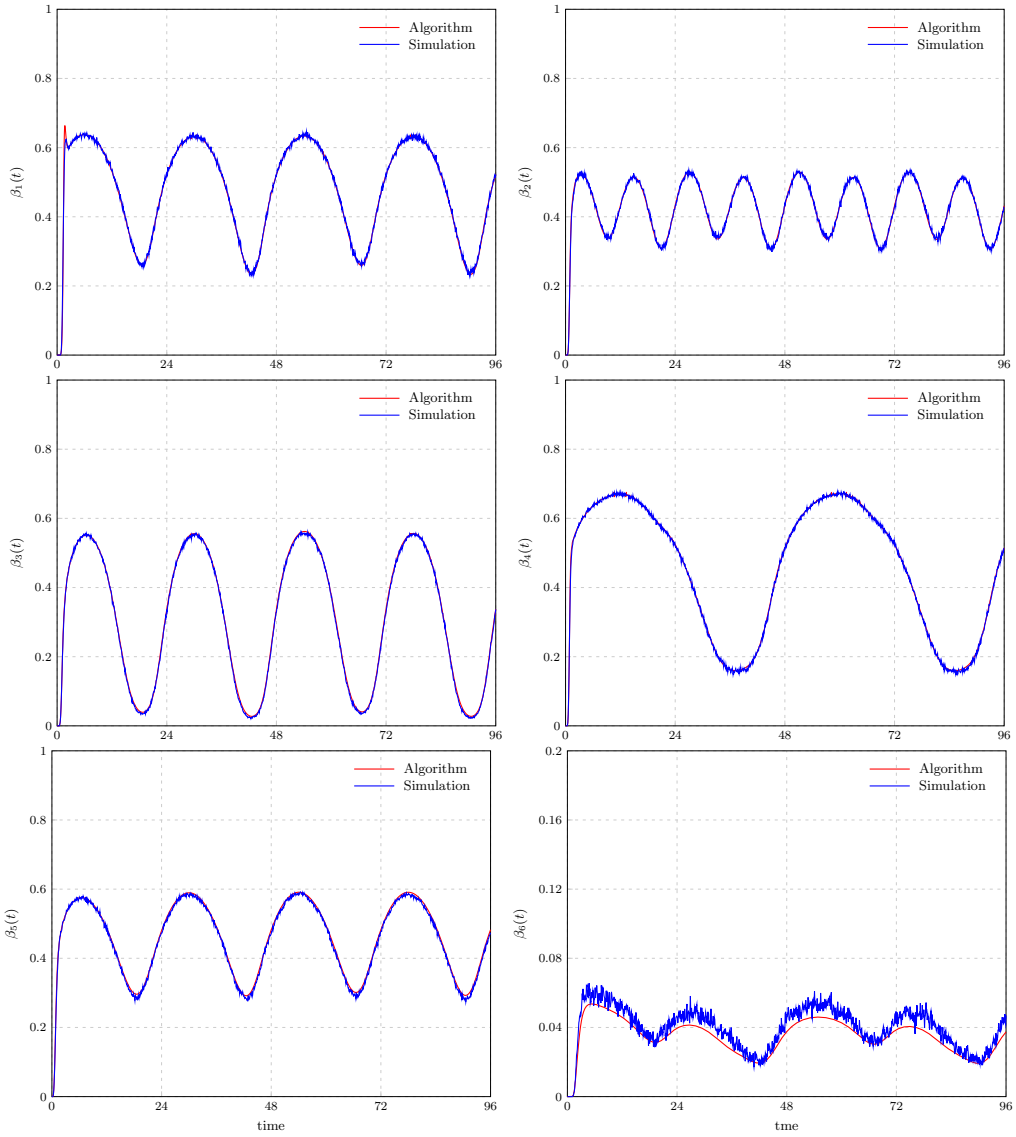

Figure 9: Blocking probabilities vs. time for the loss network example.

and its speed is almost independent of the system size.

Finally we note that throughout the paper we have used infinite-server results to derive exact equations for the time-dependent carried load in the loss systems of interest. These have then been combined with known results for stationary loss systems to give high quality approximate results for time-dependent loss systems. In the case of multi-server loss queues with a single class of customers, the stationary equation is the Erlang loss formula. In Section 2.4, we showed that this was not exact, but nevertheless seemed to encapsulate the essence of the problem. It therefore seems likely that the same approach of using time-dependent carried load derived from infinite-server systems in combination with known stationary results may also

offer an analytical way forwards in the analysis of other types of time-dependent loss systems.

Appendix A. Supplementary Material

See the online supplement to this article.

References

Abdalla, N., Boucherie, R. J., 2002. Blocking probabilities in mobile communications networks with time-varying rates and redialing subscribers. Annals of Operations Research 112 (1), 15–34.

Alnowibet, K., Perros, H., 2009. Nonstationary analysis of the loss queue and of queueing networks of loss queues. European Journal of Operational Research 196 (3), 1015–1030.

Alnowibet, K. A., Perros, H., 2006. Nonstationary analysis of circuit-switched communication networks. Performance Evaluation 63 (9-10), 892–909.

Asmussen, S., 2003. Applied Probability and Queues, 2nd Edition. Springer, Berlin.

Bekker, R., Bruin, A. M. d., 2009. Time-dependent analysis for refused admissions in clinical wards. Annals of Operations Research, to appear.

Bruin, A. M. d., Bekker, R., Zanten, L. v., Koole, G. M., 2009. Dimensioning hospital wards using Erlang loss model. Annals of Operations Research, to appear.

Davis, Jimmie, L., Massey, W. A., Whitt, W., 1995. Sensitivity to the service-time distribution in the nonstationary Erlang loss model. Management Science 41 (6), 1107–1116.

Eick, S. G., Massey, W. A., Whitt, W., 1993. The physics of the $M_t/G/\infty$ queue. Operations Research 41 (4), 731–742.

Green, L., Kolesar, P., Svoronos, A., 1991. Some effects of nonstationarity on multiserver markovian queueing systems. Operations Research 39 (3), 502–511.

Green, L. V., Kolesar, P. J., 1991. The pointwise stationary approximation for queues with nonstationary arrivals. Management Science 37 (1), 84–97.

Gross, D., Harris, C. M., 1998. Fundamentals of Queueing Theory, 3rd Edition. John Wiley & Sons, New York.

Ingolfsson, A., Akhmetshina, E., Budge, S., Li, Y., Wu, X., 2007. A survey and experimental comparison of service-level-approximation methods for nonstationary $M(t)/M/s(t)$ queueing systems with exhaustive discipline. INFORMS Journal On Computing 19 (2), 201–214.

Jagerman, D. L., 1975. Nonstationary blocking in telephone traffic. Bell Systems Technical Journal 54, 625–661.

Jennings, O., Massey, W., 1997. A modified offered load approximation for nonstationary circuit switched networks. Telecommunication Systems 7 (1), 229–251.

Kaufman, J., 1981. Blocking in a shared resource environment. Communications, IEEE Transactions on Communications 29 (10), 1474–1481.

Massey, W. A., Whitt, W., 1993. Networks of infinite-server queues with nonstationary Poisson input. Queueing Systems 13 (1), 183–250.

Massey, W. A., Whitt, W., 1994. An analysis of the modified offered-load approximation for the nonstationary Erlang loss model. The Annals of Applied Probability 4 (4), 1145–1160.

Massey, W. A., Whitt, W., 1996. Stationary-process approximations for the nonstationary Erlang loss model. Operations Research 44 (6), 976.

Shampine, L. F., Reichelt, M. W., 1997. The MATLAB ODE suite. SIAM Journal on Scientific Computing 18 (1), 1–22.

Whitt, W., 1991. The pointwise stationary approximation for $M_t/M_t/s$ queues is asymptotically correct as the rates increase. Management Science 37 (3), 307–314.

Wolff, R. W., 1988. Stochastic Modeling and the Theory of Queues. Prentice Hall, New Jersey.